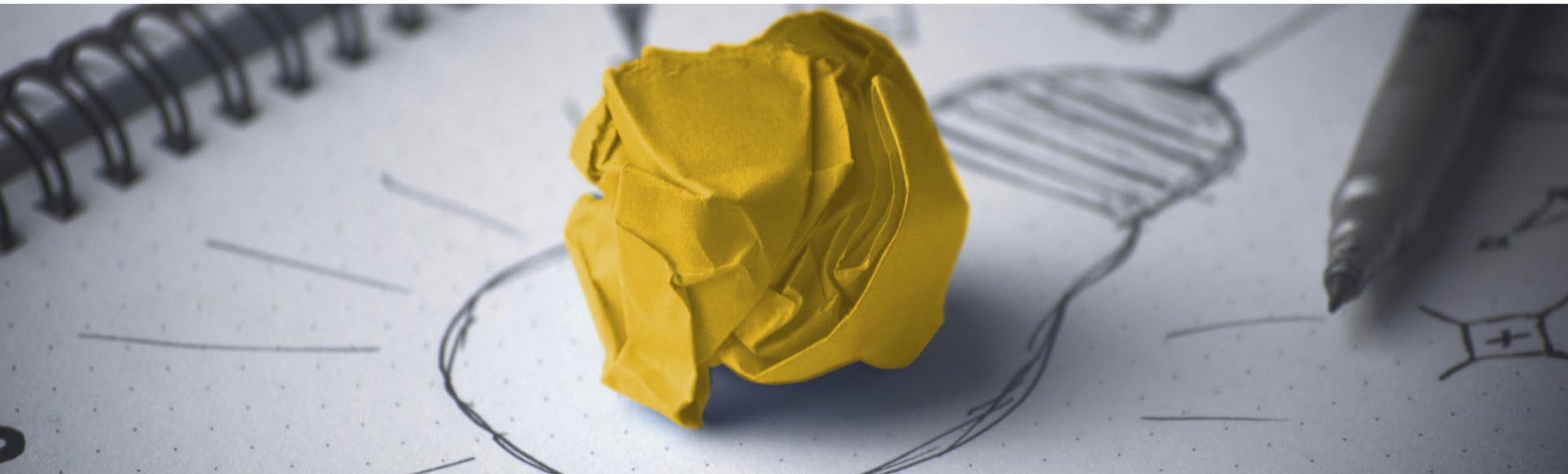


Road Show

Case Study



- Praxisbeispiel
- Change Detection Zone
- Angular Elements
- ngrx (Augury)
- Cordova / Ionic
- NativeScript
- Angular Universal
- AngularJS => Angular Bridge



Die 2BIT GmbH vereinfacht die technologische Nutzung der Enduser in der Sprache des Kunden auf allen Geräten.

PWA made easy

```
D:\git\2school-angular-advanced>ng add @angular/pwa
Installing packages for tooling via npm.
Installed packages for tooling via npm.
CREATE projects/todo-mvc-s/ngsw-config.json (624 bytes)
CREATE projects/todo-mvc-s/src/manifest.webmanifest (1344 bytes)
CREATE projects/todo-mvc-s/src/assets/icons/icon-128x128.png (1253 bytes)
CREATE projects/todo-mvc-s/src/assets/icons/icon-144x144.png (1394 bytes)
CREATE projects/todo-mvc-s/src/assets/icons/icon-152x152.png (1427 bytes)
CREATE projects/todo-mvc-s/src/assets/icons/icon-192x192.png (1790 bytes)
CREATE projects/todo-mvc-s/src/assets/icons/icon-384x384.png (3557 bytes)
CREATE projects/todo-mvc-s/src/assets/icons/icon-512x512.png (5008 bytes)
CREATE projects/todo-mvc-s/src/assets/icons/icon-72x72.png (792 bytes)
CREATE projects/todo-mvc-s/src/assets/icons/icon-96x96.png (958 bytes)
UPDATE angular.json (4184 bytes)
UPDATE package.json (1559 bytes)
UPDATE projects/todo-mvc-s/src/app/app.module.ts (1331 bytes)
UPDATE projects/todo-mvc-s/src/index.html (573 bytes)
✓ Packages installed successfully.
```

```
ng add @angular/pwa --project
```

```
ng build --prod
```

Run PWA Project

- `npm install --global http-server`

```
http-server -p 8080 -c-1 dist/<project-name>
```

Service Worker - JSON

The screenshot displays a web application interface on the left and Chrome DevTools on the right. The application, titled "2BIT Angular examples", features a "Todo example" section with "MVC+S architecture". It includes a form to "Add a new Item" with a text input labeled "Todo Beschreibung" and an "Add" button. Below the form, there is a list of todos: "Todo" (with a notepad icon) and "Done" (with a green checkmark icon).

The Chrome DevTools interface on the right shows the "Service Workers" tab. It lists a service worker for "http://localhost:8080/" with source "ngsw-worker.js", received on "6.9.2020, 12:39:35", and status "#5011 activated and is running". The "Console" tab below shows several error messages:

- returned response 504 Gateway Timeout
at LazyAssetGroup.<anonymous> (ngsw-worker.js:656)
at Generator.next (<anonymous>)
at fulfilled (ngsw-worker.js:196)
- Failed to load resource: net::ERR_FAILED favicon.png:1
- Failed to load resource: the server responded with a status of 404 (Not Found) todo:1

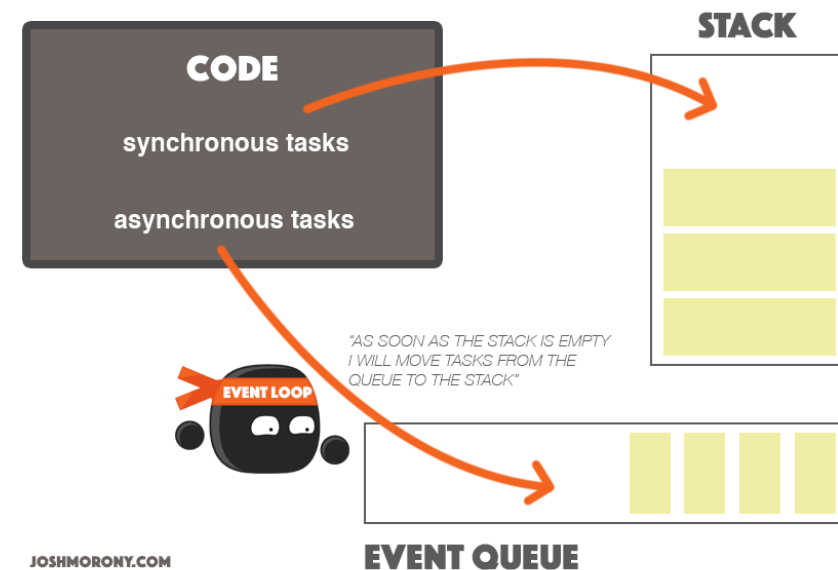
Configuration prefetch

```
{...} ngsw-config.json X
projects > todo-mvc-s > {...} ngsw-config.json > [ ] assetGroups >
1 {
2   "$schema": "../../node_modules/@angular/sc
3   "index": "/index.html",
4   "assetGroups": [
5     {
6       "name": "app",
7       "installMode": "prefetch",
8       "resources": {
9         "files": [
10          "/favicon.ico",
11          "/index.html",
12          "/manifest.webmanifest",
13          "/*.css",
14          "/*.js",
15          "/assets/mock-data/todos.json"
16        ]
17      }
18    }, {
19      "name": "assets",
20      "installMode": "lazy",
21      "updateMode": "prefetch",
22      "resources": {
23        "files": [
24          "/assets/**",
25          "/*.(eot|svg|cur|jpg|png|webp|gif|
26        ]
27      }
28    }
29  ]
}
```

```
export interface DataGroup {
  name: string;
  urls: string[];
  version?: number;
  cacheConfig: {
    maxSize: number;
    maxAge: string;
    timeout?: string;
    strategy?: 'freshness' | 'performance';
  };
  cacheQueryOptions?: {
    ignoreSearch?: boolean;
  };
}
```

Change Detection – Zones

- Angular läuft in ihrer eigenen Zone
 - NgZone
 - Change Detection
 - Monkey Patches System
 - setTimeout()
 - addEventListener()



Outside NgZone

```
increaseProgress(doneCallback: () => void) {
  this.progress += 1;
  console.log(`Current progress: ${this.progress}%`);

  if (this.progress < 100) {
    window.setTimeout(() => {
      this.increaseProgress(doneCallback);
    }, 10);
  } else {
    doneCallback();
  }
}
```

```
processOutsideAngularZone() {
  this.progress = 0;
  this.zone.runOutsideAngular(() => {
    this.increaseProgress(() => {
      this.zone.run(() => {
        console.log('Outside Done!');
      });
    });
  });
}
```

```
@Component({
  selector: 'progress-bar',
  template: `
    <h3>Progress: {{progress}}</h3>
    <button (click)="processWithinAngularZone()">
      Process within Angular zone
    </button>
  `
})
class ProgressBar {

  progress: number = 0;

  constructor(private zone: NgZone) {}

  processWithinAngularZone() {
    this.progress = 0;
    this.increaseProgress(() => console.log('Done!'));
  }
}
```

```
this.zone.runOutsideAngular(() => {

  // do something

  // reenter Angular zone
  this.zone.run(() => {
    // back in Angular town
  });
});
```


- Angular ist ein komplettes Frontend Framework, was aber wenn wir nur einzelne komplexe Komponenten in einer Webseite einbinden möchten?



Angular Elements

WEB COMPONENTS



HTML
IMPORTS



SHADOW
DOM



TEMPLATES



CUSTOM
ELEMENTS

Angular Elements

- Custom Elements ermöglichen das registrieren von eigenen Elementen die der Browser nicht native kennt
- Ein eigenes Element kann über die document/window API registriert werden
 - `var myElement = document.registerElement('my-element');` (v0)
 - `window.customElements.define('my-element', MyElement);` (v1)

Angular Elements

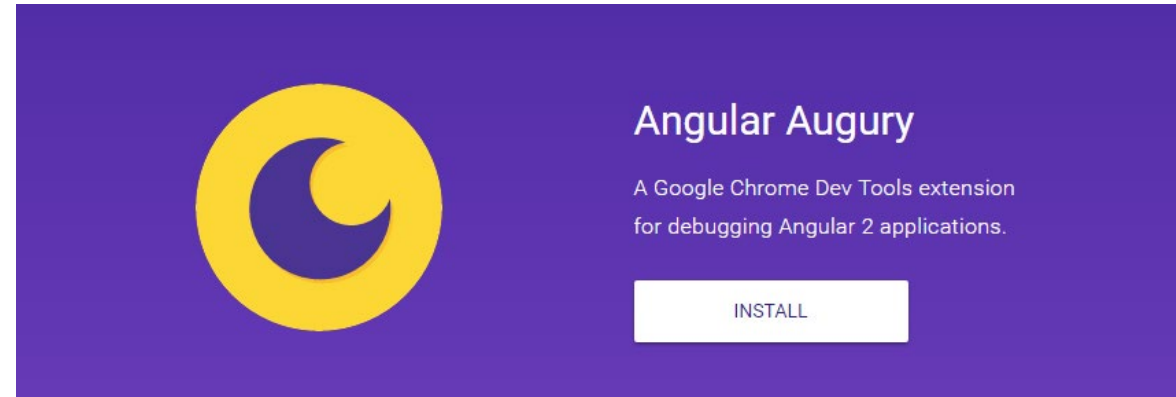
- Anschliessend weiss der Browser wie folgendes HTML zu interpretieren ist
 - `<my-element></my-element>`

<https://angular.io/guide/elements>

Angular Elements

- WebComponents die als CustomElements registriert werden bestehen wie jede HTML Komponente aus Inputs und Outputs
- Angular Componentes haben die selben eigenschaften, weswegen diese mit etwas Hilfe von Angular Elements relativ einfach zu Custom Elements kompiliert werden können
- Hier setzt Angular Elements an, welches in einer ersten Version mit Angular 6 releast wurde, aber noch nicht Feature complete ist

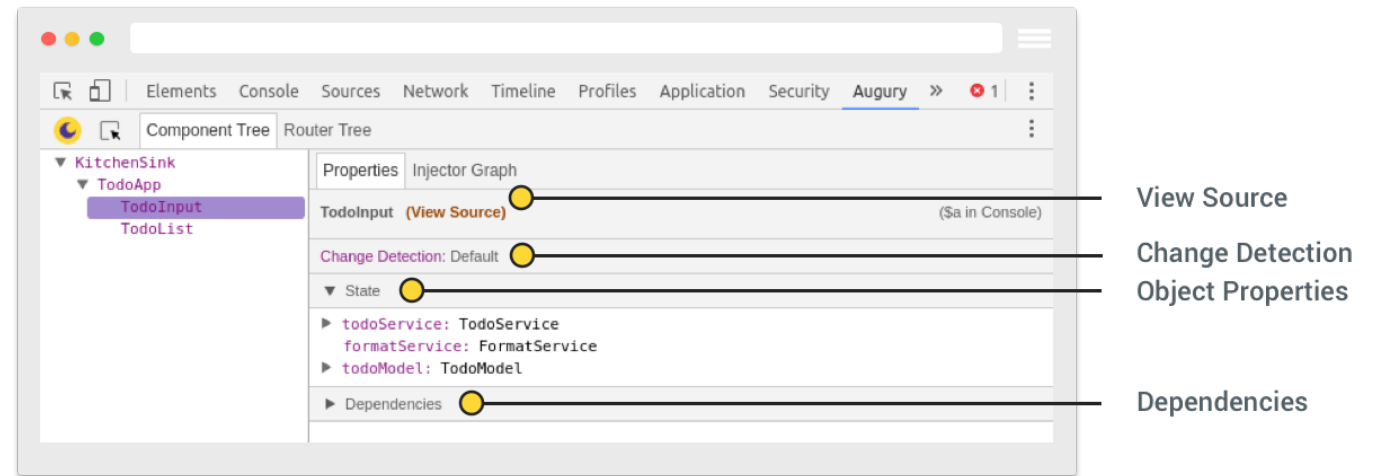
Augury



- <https://augury.angular.io/>
- <https://augury.angular.io/pages/guides/>

- Visual

- Component Trees
 - Visual Debbing
- Immediate insight
 - application structure
 - change detection
 - performance characteristics



- Angular Universal bietet verschiedene Features für Angular auf der Serverseite
 - Serverseitiges Rendering der Startseite um die Anzeige Geschwindigkeit der Landingpage zu verbessern
 - Verbesserungen für die Indexierung durch Search Engines
 - Bereitstellen einer passenden Site Preview
 - Für andere Seiten die Preview Images darstellen (Facebook, Twitter...)

<https://universal.angular.io/overview/>

<https://angular.io/guide/universal>



Angular Universal



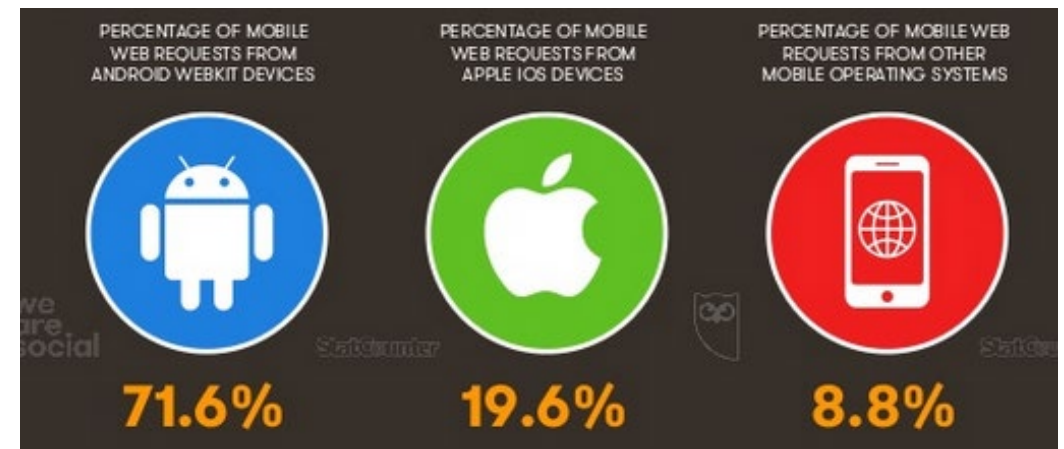
- Browser receives initial payload from server
- User sees server view
- Preboot creates hidden div that will be used for client bootstrap and starts recording events
- Browser makes async requests for additional assets (i.e. images, JS, CSS, etc.)
- Once external resources loaded, Angular client bootstrapping begins
- Client view rendered to the hidden div created by Preboot
- Bootstrap complete, so Angular client calls `preboot.done()`
- Preboot events replayed in order to adjust the application state to reflect changes made by the user before Angular bootstrapped (i.e. typing in textbox, clicking button, etc.)
- Preboot switches the hidden client view div for the visible server view div
- Finally, Preboot performs some cleanup on the visible client view including setting focus

- Können Webtechnologien auch für Apps eingesetzt werden?
- Möglichkeiten dazu?
- Was sind potentielle Probleme damit?

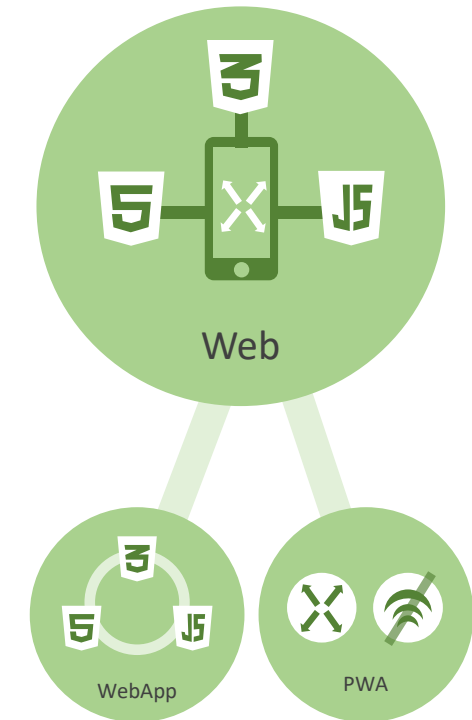
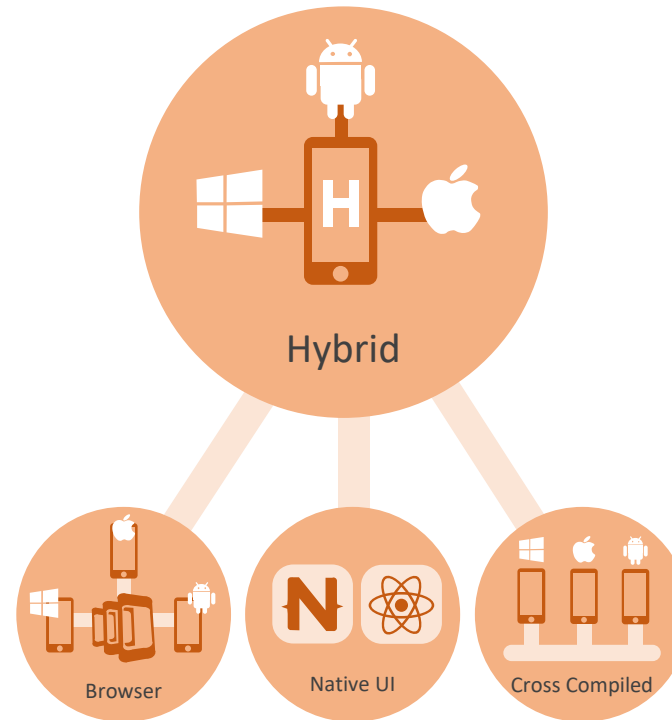
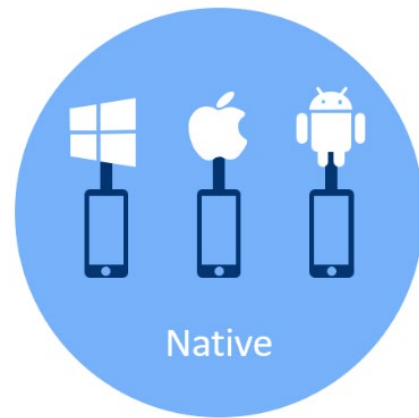


- Herausforderungen beim bedienen von verschiedenen Plattformen
 - Parallele Native Entwicklung (Java, Objective-C, .NET)
 - Hybride Entwicklung
 - Web Applikation mit Zugriff auf Geräte über 3rd Parties
 - Cross Development (Xamarin, NativeScript)

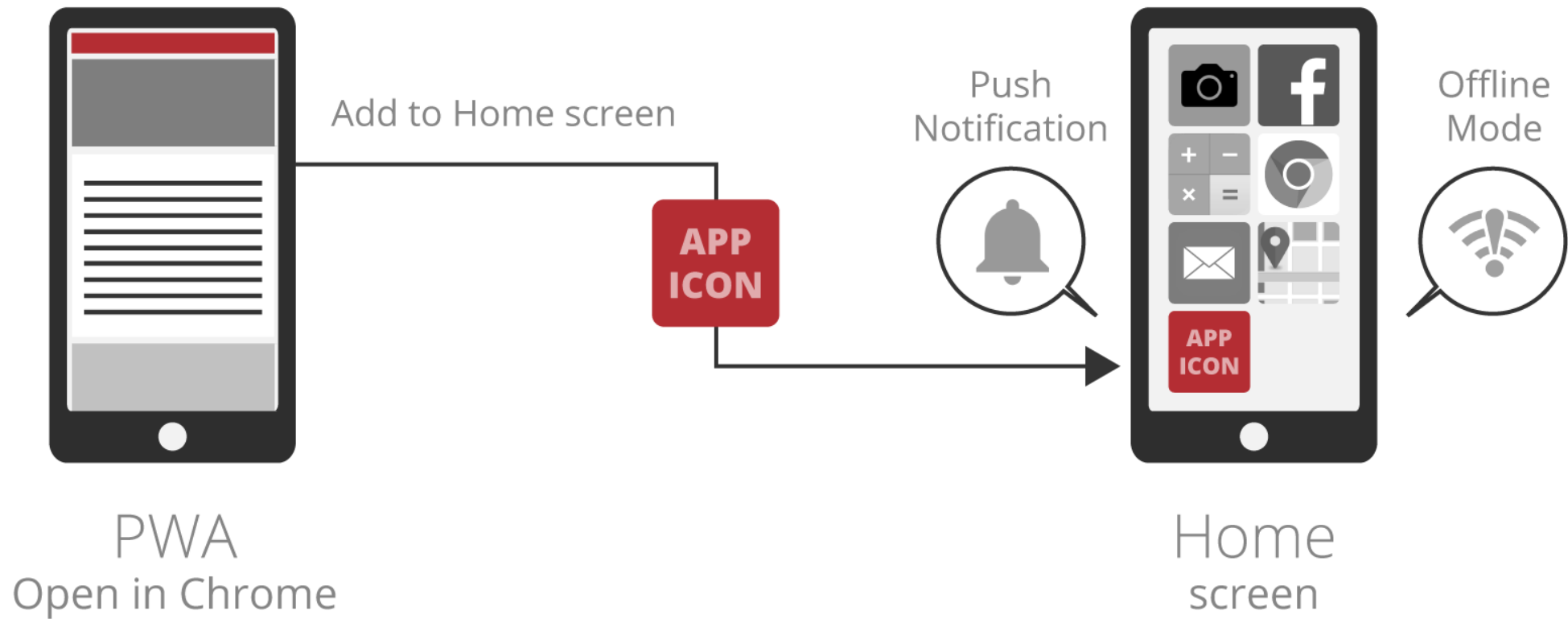
- Es gibt heutzutage primär zwei verschiedene Plattformen
 - IOS
 - Android
- Viele verschiedene Gerätegrößen
- Unterschiedliche Gerätearchitekturen

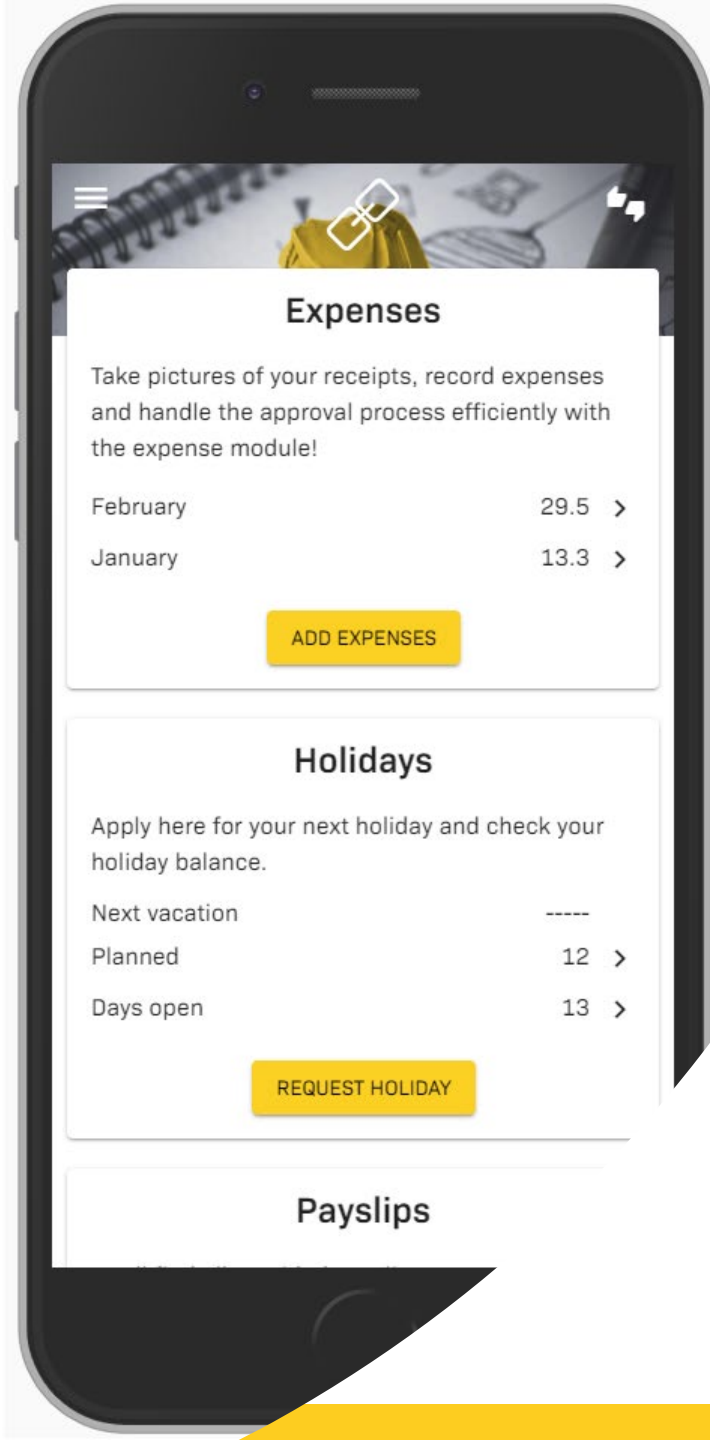


Arten der Mobile-Entwicklung

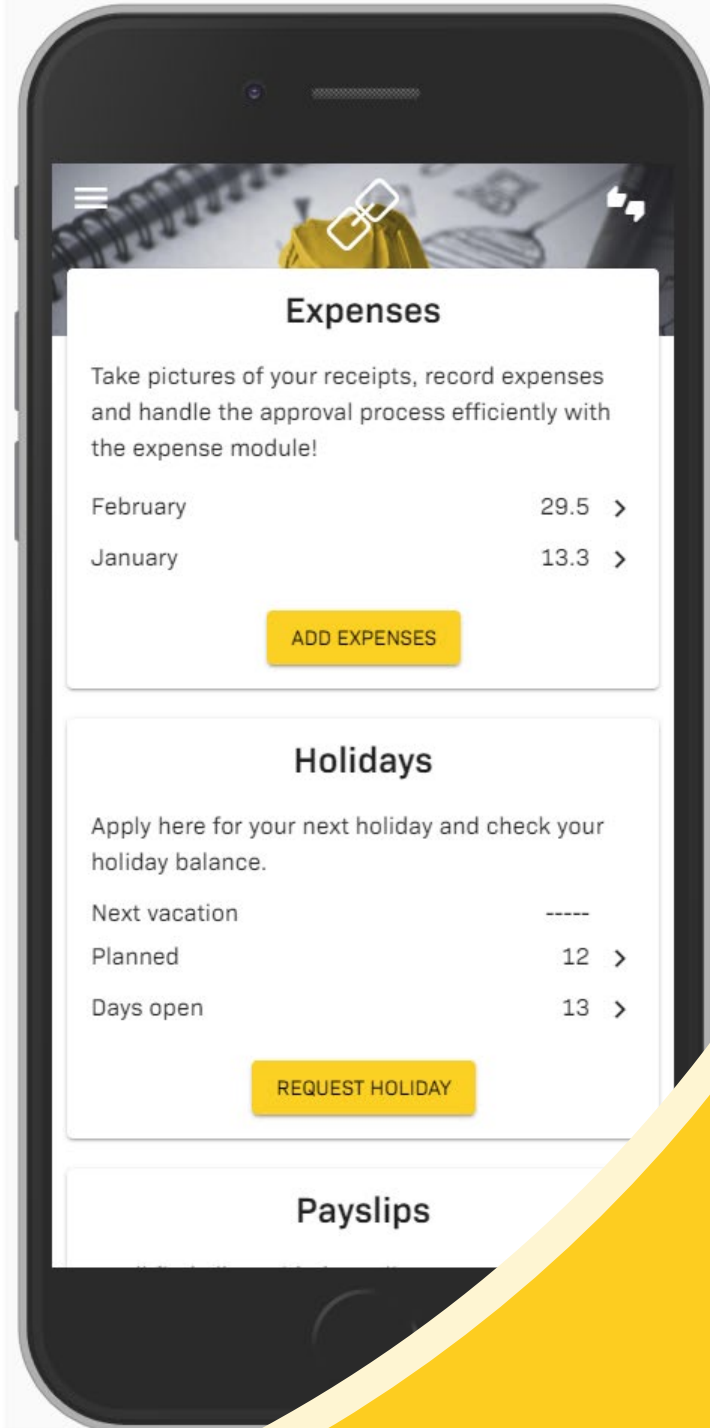


Progressive Web App





PWA in Action
2gethr.2bitcloud.ch



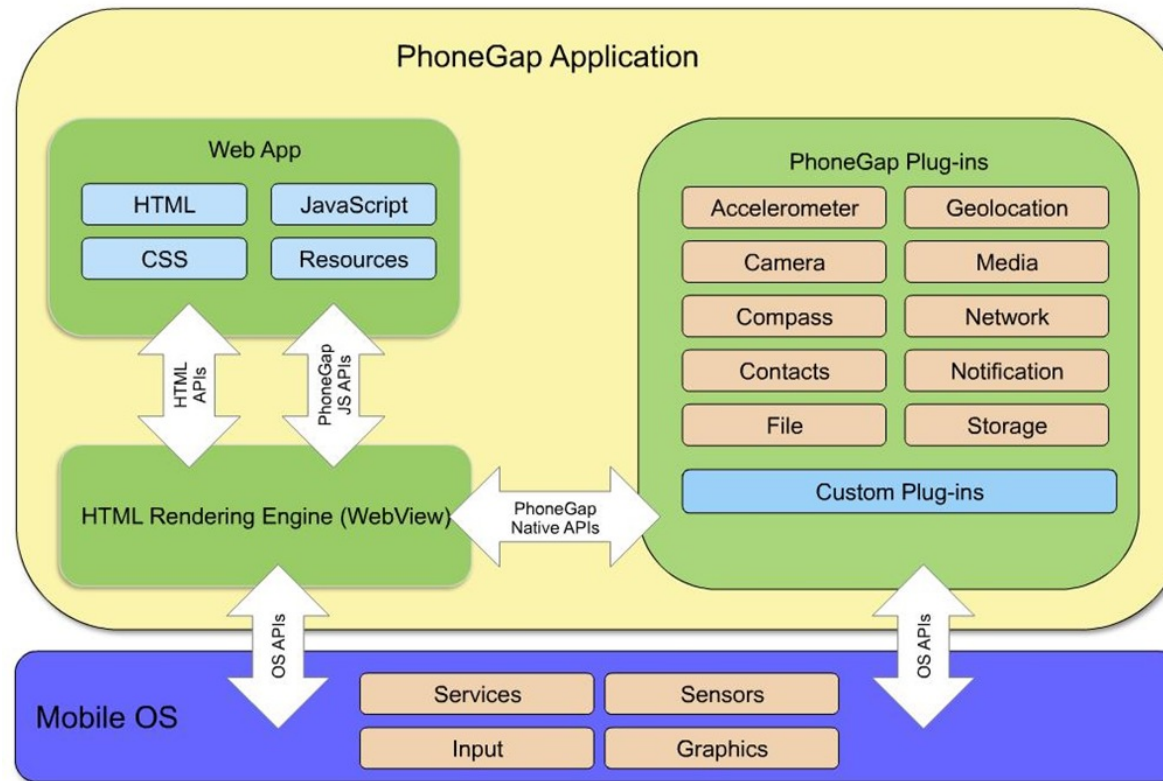
PWA in Action
2gethr.2bitcloud.ch

- PhoneGap bietet im Core schon verschiedenste Funktionalitäten an, es gibt aber zusätzlich noch unzählige (teilweise wieder gerätespezifische) Plugins

	iPhone / iPhone 3G	iPhone 3GS and newer	Android	Blackberry OS 6.0+	Blackberry 10	Windows Phone 8	Ubuntu	Firefox OS
Accelerometer	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓
Compass	X	✓	✓	X	✓	✓	✓	✓
Contacts	✓	✓	✓	✓	✓	✓	✓	✓
File	✓	✓	✓	✓	✓	✓	✓	X
Geolocation	✓	✓	✓	✓	✓	✓	✓	✓
Media	✓	✓	✓	X	✓	✓	✓	X
Network	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Alert)	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Sound)	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Vibration)	✓	✓	✓	✓	✓	✓	✓	✓
Storage	✓	✓	✓	✓	✓	✓	✓	✓

Cordova/PhoneGap/Capacitor

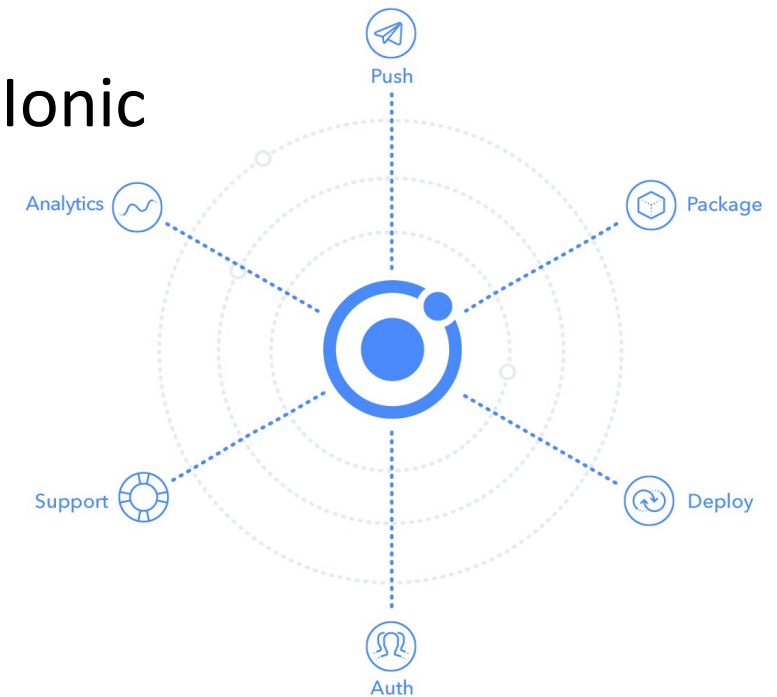
PhoneGap Architecture



- Ionic ist ein Framework welches auf Cordova aufsetzt um Apps mit der Hilfe von Webtechnologien zu entwickeln
 - Das Ionic Framework gibt es schon relativ lang und ist sowohl für AngularJS als auch für Angular verfügbar
 - Die Version für Angular befindet sich jedoch noch in einem Beta-Stadium!
- Ionic bietet bereits viele Standardfunktionalitäten und grafische Elemente an



- Neben Framework bietet Ionic auch verschiedenste Infrastrukturdienstleistungen an, welche jedoch nicht alle kostenlos sind
- Hier einen kurzen Überblick, was für Controls Ionic bereits anbietet
 - <http://ionicframework.com/docs/components>



NativeScript

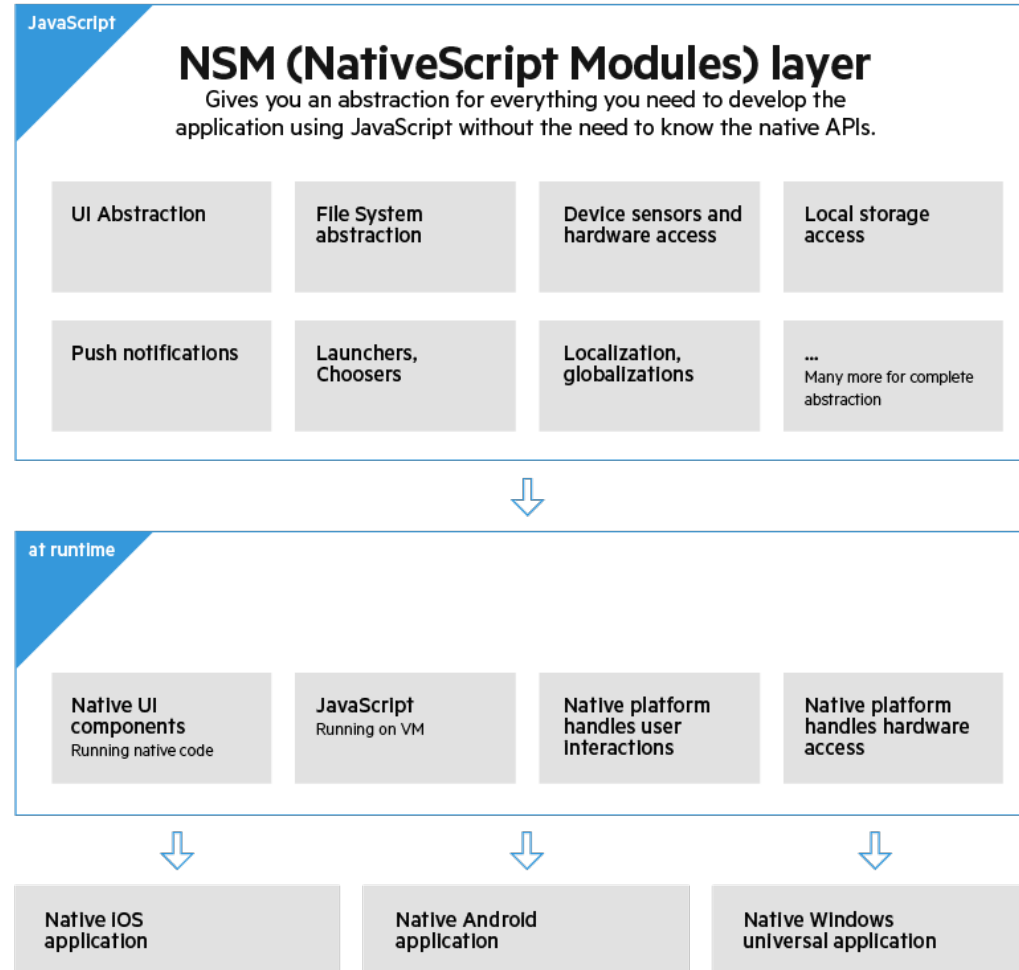


NativeScript

- NativeScript gibt es in verschiedenen Varianten und ist komplett Open Source
 - JavaScript Variante
 - Angular Variante
- NativeScript gibt Angular Entwicklern also die Möglichkeit mit Angular «NATIVE» Apps für die beiden Plattformen iOS und Android zu entwickeln
- Das heisst das Know-how aus der Web Entwicklung kann direkt wieder verwendet werden für die App Entwicklung



NativeScript



NativeScript



NativeScript

- Die aktuelle Version von NativeScript ist 4.2
- Die erste finale Version von NativeScript mit Angular ist im Juni 2016 mit der Version 2.1 erschienen
- Ein Beispiel wie hoeher Code Reuse erreicht werden kann zwischen der Web App und mobile App findet man hier

<https://blog.angular.io/apps-that-work-natively-on-the-web-and-mobile-9b26852495e7>

NativeScript Demo



<https://play.nativescript.org/>

NativeScript

- Was sind Vor- und Nachteile von Cross Platform Development mit einer Technologie wie NativeScript gegenüber einer App mit reinen Webtechnologien?
- Wie schafft es Native Script aus einer Angular Applikation eine Native App zu bilden?

